IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Appl. No.	:	09/877,928	Confirmation No.:	1359
Applicant	:	Jensen et al.		
Filed	:	June 7, 2001		
TC/A.U.	:	2153		
Examiner	:	Reilly, Sean M.		
Docket No.	:	1020.P10678		

Mail Stop Appeal Brief
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF

This Appeal Brief is in furtherance of the Notice of Appeal filed on November 28, 2006 and the Notification of Non-Compliant Appeal Brief mailed September 24, 2007.

The Appeal Brief contains the following sections in the order set forth below:

I. REAL PARTY IN INTEREST
II. RELATED APPEALS AND INTERFERENCES
III. STATUS OF CLAIMS
IV. STATUS OF AMENDMENTS
V. SUMMARY OF THE CLAIMED SUBJECT MATTER
VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL
VII. ARGUMENT
VIII. CLAIMS APPENDIX
IX. EVIDENCE APPENDIX
X. RELATED PROCEEDINGS APPENDIX

I. REAL PARTY IN INTEREST

The real party in interest in this appeal is Intel Corporation, as the Assignee of record.

II. RELATED APPEALS AND INTERFERENCES

There are no other appeals or interferences that will directly affect, or be affected by, or have a bearing on the decision of the Board in the pending appeal.

III. STATUS OF CLAIMS

Claims originally filed: 1-18

Claims canceled: None

Claims withdrawn from consideration: None

Claims allowed: None

Claims objected to: None

Claims rejected: 1-18

Claims on appeal: 1-18

IV. STATUS OF AMENDMENTS

The Amendment filed on September 28, 2006, subsequent to the Final Rejection, has been entered.

V. SUMMARY OF THE CLAIMED SUBJECT MATTER

The following is a concise explanation of the subject matter defined in each of the independent claims involved in the appeal. Independent claims 1, 8, and 15 are fully supported, concisely explained, and map to the specification and drawings.

Independent claim 1 maps to the specification and to the drawings as follows.

Claim 1	Specification and Drawings
1. A method to retrieve information, comprising:	An improved method and apparatus to retrieve information over a network is described. See **ABSTRACT**.
receiving a first request for information from a client at a network node over a first connection;	A first request for information (e.g., request for a web page using an HTTP "Get" command) is received over a first connection at block 302. The first connection may be a connection between a client computer 102 and an access device 106. See **Page 10, lines 9-13** and **FIG. 3.**
establishing a second connection to retrieve said information;	The information may be retrieved over a second connection at block 304. The second connection may be a connection between an access device 106 and a server 108 containing the requested information. See **Page 10, lines 13-16** and **FIG. 3.** The access device 106 (e.g., web switch) may set up a TCP connection toward the selected server, and passes the URL request to the server over the opened connection. See **Page 12, lines 13-14** and **FIG. 4**.
detecting that said first connection is terminated prior to retrieval of said information over said second connection;	The access device 106 may detect that the first connection has been terminated at block 306. The first connection may be terminated for various reasons, such as intentionally by the user via a cancellation or refresh request sent from a browser at the client computer 102, a break in the physical or logical connection between the client computer 102 and access device 106, and so forth. See **Page 10, lines 16-20** and **FIG. 3.**
retrieving said information over said second connection;	The information may be retrieved over a second connection at block 304. The

	second connection may be a connection between an access device 106 and a server 108 containing the requested information. See **Page 10, lines 13-16** and **FIG. 3.**
receiving a second request at the network node for said information over a third connection;	The access device 106 may receive a second request for the information over a third connection at block 308. The third connection may comprise, for example, a new connection between the client computer 102 and the access device 106 to replace the terminated first connection. See **Page 10, lines 20-23** and **FIG. 3.**
determining whether said second request matches said first request, including whether said second request is from said client;	The access device 106 may determine whether the second request matches the first request at block 310. See **Page 10, line 23 – Page 11, line 1** and **FIG. 3.** After retrieving the information at block 304, the access device 106 may store the information in memory (e.g., memory 210, disk storage 218 of FIG. 2). For example, the access device 106 may store a first source address, a first information address, and the information in an information table. The access device 106 may receive a second request (e.g., second source address and a second information address) at block 308. The access device 106 may search the information table to determine whether the second source address matches the first source address, and the first information address matches the second information address. See **Page 11, lines 4-13** and **FIG. 3**.
sending said information over said third connection in accordance with said determination; and	The access device 106 may send the information over the third connection at block 312 in accordance with the determination at block 310. See **Page 10, lines 1-3** and **FIG. 3.** If both match, the access device 106 may retrieve the information corresponding to the first source address and first information address, and send the information to the client computer over the third connection. See **Page 11, lines 13-15** and **FIG. 3**.

deleting said information at the network node upon delivery of said information to said client.	The access device 106 (e.g., web switch) may delete the information (e.g., web page) and corresponding entries from its information table once the information is sent to the client computer 102. See **Page 13, lines 15-16**.

Independent claim 8 maps to the specification and to the drawings as follows.

Claim 8	**Specification and Figures**
8. A method to retrieve information in a network, comprising:	An improved method and apparatus to retrieve information over a network is described. See **ABSTRACT**. Network 100 may comprise a client computer 102, a network 104, an access device 106 and a server 108. See **Page 7, lines 1- 2** and **FIG. 1**.
receiving a first request for information from a first network node at a second network node over a first connection;	A first request for information (e.g., request for a web page using an HTTP "Get" command) is received over a first connection at block 302. The first connection may be a connection between a client computer 102 and an access device 106. See **Page 10, lines 9-13** and **FIG. 3.**
sending said first request over a second connection to a third network node;	The information may be retrieved over a second connection at block 304. The second connection may be a connection between an access device 106 and a server 108 containing the requested information. See **Page 10, lines 13-16** and **FIG. 3.** The access device 106 (e.g., web switch) may set up a TCP connection toward the selected server, and passes the URL request to the server over the opened connection. See **Page 12, lines 13-14** and **FIG. 4**.
receiving a notice that said first connection is terminated prior to retrieval of said information over said second connection;	The access device 106 may detect that the first connection has been terminated at block 306. The first connection may be terminated for various reasons, such as intentionally by the user via a cancellation or refresh request sent from a browser at the client computer 102, a break in the physical or logical connection between the

	client computer 102 and access device 106, and so forth. See **Page 10, lines 16-20** and **FIG. 3.**
receiving said information over said second connection at said second network node;	The information may be retrieved over a second connection at block 304. The second connection may be a connection between an access device 106 and a server 108 containing the requested information. See **Page 10, lines 13-16** and **FIG. 3.**
receiving a second request for said information over a third connection at said second network node;	The access device 106 may receive a second request for the information over a third connection at block 308. The third connection may comprise, for example, a new connection between the client computer 102 and the access device 106 to replace the terminated first connection. See **Page 10, lines 20-23** and **FIG. 3.**
determining whether said second request matches said first request, including whether said second request is from said first network node;	The access device 106 may determine whether the second request matches the first request at block 310. See **Page 10, line 23 – Page 11, line 1** and **FIG. 3.** After retrieving the information at block 304, the access device 106 may store the information in memory (e.g., memory 210, disk storage 218 of FIG. 2). For example, the access device 106 may store a first source address, a first information address, and the information in an information table. The access device 106 may receive a second request (e.g., second source address and a second information address) at block 308. The access device 106 may search the information table to determine whether the second source address matches the first source address, and the first information address matches the second information address. See **Page 11, lines 4-13** and **FIG. 3**.
sending said information over said third connection to said first network node; and	The access device 106 may send the information over the third connection at block 312 in accordance with the determination at block 310. See **Page 10, lines 1-3** and **FIG. 3.** If both match, the access device 106 may retrieve the information corresponding to

	the first source address and first information address, and send the information to the client computer over the third connection. See **Page 11, lines 13-15** and **FIG. 3**.
deleting said information at the second network node upon delivery of said information to said first network node.	The access device 106 (e.g., web switch) may delete the information (e.g., web page) and corresponding entries from its information table once the information is sent to the client computer 102. See **Page 13, lines 15-16**.

Independent claim 15 maps to the specification and to the drawings as follows.

Claim 15	**Specification and Figures**
15. An article comprising: a storage medium; said storage medium including stored instructions that, when executed by a processor, result in	System 200 may be representative of access device 106. System 200 may comprise a processor 202, an input/output (I/O) adapter 204, an operator interface 206, a memory 210 and a disk storage 218. Memory 210 may store computer program instructions and data. The term "program instructions" may include computer code segments comprising words, values and symbols from a predefined computer language that, when placed in combination according to a predefined manner or syntax, cause a processor to perform a certain function. See **Page 7, lines 5-12** and **FIG. 2**. FIG. 3 is a block flow diagram of the programming logic performed by an information management component in accordance with one embodiment of the invention. See **Page 10, lines 7-9** and **FIG. 3.**
receiving a first request for information from a client at a network node over a first connection,	A first request for information (e.g., request for a web page using an HTTP "Get" command) is received over a first connection at block 302. The first connection may be a connection between a client computer 102 and an access device 106. See **Page 10, lines 9-13** and **FIG. 3.**

establishing a second connection to retrieve said information,	The information may be retrieved over a second connection at block 304. The second connection may be a connection between an access device 106 and a server 108 containing the requested information. See **Page 10, lines 13-16** and **FIG. 3.** The access device 106 (e.g., web switch) may set up a TCP connection toward the selected server, and passes the URL request to the server over the opened connection. See **Page 12, lines 13-14** and **FIG. 4**.
detecting that said first connection is terminated prior to retrieval of said information over said second connection,	The access device 106 may detect that the first connection has been terminated at block 306. The first connection may be terminated for various reasons, such as intentionally by the user via a cancellation or refresh request sent from a browser at the client computer 102, a break in the physical or logical connection between the client computer 102 and access device 106, and so forth. See **Page 10, lines 16-20** and **FIG. 3.**
retrieving said information over said second connection,	The information may be retrieved over a second connection at block 304. The second connection may be a connection between an access device 106 and a server 108 containing the requested information. See **Page 10, lines 13-16** and **FIG. 3.**
receiving a second request for said information at the network node over a third connection,	The access device 106 may receive a second request for the information over a third connection at block 308. The third connection may comprise, for example, a new connection between the client computer 102 and the access device 106 to replace the terminated first connection. See **Page 10, lines 20-23** and **FIG. 3.**
determining whether said second request matches said first request including whether said second request is from said client,	The access device 106 may determine whether the second request matches the first request at block 310. See **Page 10, line 23 – Page 11, line 1** and **FIG. 3.** After retrieving the information at block 304, the access device 106 may store the information in memory (e.g., memory 210, disk storage 218 of FIG. 2). For example,

	the access device 106 may store a first source address, a first information address, and the information in an information table. The access device 106 may receive a second request (e.g., second source address and a second information address) at block 308. The access device 106 may search the information table to determine whether the second source address matches the first source address, and the first information address matches the second information address. See **Page 11, lines 4-13** and **FIG. 3**.
sending said information over said third connection in accordance with said determination, and	The access device 106 may send the information over the third connection at block 312 in accordance with the determination at block 310. See **Page 10, lines 1-3** and **FIG. 3.** If both match, the access device 106 may retrieve the information corresponding to the first source address and first information address, and send the information to the client computer over the third connection. See **Page 11, lines 13-15** and **FIG. 3**.
deleting said information at the network node upon delivery of said information to said client.	The access device 106 (e.g., web switch) may delete the information (e.g., web page) and corresponding entries from its information table once the information is sent to the client computer 102. See **Page 13, lines 15-16**.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Whether claims 1, 2-5, 7, 8-12, and 14-16 are unpatentable under 35 U.S.C. § 103(a) over U.S. Patent No. 6,421,733 to Tso et al. ("Tso") and WinRoute Pro 3.0 User's Manual ("WinRoute") and U.S. Patent No. 5,991,306 to Burns ("Burns").

Whether claims 6 and 13 are unpatentable over Tso and WinRoute and Burns in view of well-known Internet standards.

VII. ARGUMENT

Claims 1, 2-5, 7, 8-12, and 14-16 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 6,421,733 to Tso et al. ("Tso") and WinRoute Pro 3.0 User's Manual ("WinRoute") and U.S. Patent No. 5,991,306 to Burns ("Burns").

Claims 6 and 13 stand rejected as being unpatentable under 35 U.S.C. § 103(a) over Tso and WinRoute and Burns in view of well-known Internet standards.

Among its other elements, independent claim 1 recites "deleting said information at the network node upon delivery of said information to said client." Among its other elements, independent claim 8 recites "deleting said information at the second network node upon delivery of said information to said first network node." Among it other elements, independent claim 15 recites "deleting said information at the network node upon delivery of said information to said client."

In the Final Office Action, independent claims 1, 8, and 15 were rejected by the Examiner for substantially the same reasons. Namely, the Examiner asserts that while Tso and WinRoute do not disclose deleting information upon delivery to a client, it would have been obvious to implement a cache deletion policy disclosed by Burns within

a Tso/WinRoute combined system in such as way as to delete requested information from a proxy cache upon delivery of the information to a client. The Examiner's rationale for making such combination and modification is premised on managing the capacity limitations of a cache memory by deleting content as soon as possible to more efficiently utilize the cache memory.

Specifically, on pages 5-6 of the Final Office Action, the Examiner stated as follows:

> Neither Tso nor WinRoute disclosed deleting said information upon delivery of said information to said client. Nonetheless it was widely known in the art at the time of Applicant's invention that proxy caches such as those employed by both Tso and WinRoute have limited capacity, as evidenced by at least Burns. In and analogous proxy cache system (see inter alia Figure 2) Burns disclosed, "deletion policies are a function of… the constraints imposed by capacity limitations of the cache memory" (see Burns Col 11, lines 15-19). Thus, it would have been obvious to one of ordinary skill in the art at the time of Applicant's invention to implement a cache deletion policy in the combined Tso and WinRoute system which deletes requested information from the proxy cache upon delivery of said information to a client, in order to manage the capacity limitations of the cache memory by deleting content as soon as possible and thus more efficiently utilizing the cache memory. The cache memory is more efficiently utilized since more users can have their requests cached when cache memory is freed up as soon as possible. In addition by caching more users' requests the system is able to more efficiently respond to the requests.

In the Advisory Action, the Examiner stated as follows:

> Applicant's arguments are not persuasive. Notably, Examiner maintains that it would have been obvious to modify WinRoute's system to delete requested data from the proxy after the data is sent to the requesting client and no longer needed at the proxy in order to free up capacity in the cache memory. Examiner also notes that the fact that a system is a caching system does not imply that cached data

> will not be deleted immediately after use or that cached data must be stored for long periods of time. As was widely known in the art at the time of Applicant's invention and further disclosed by Burns, data may be deleted based on TTL values (time to live), subscriber patterns, or constraints on the size of the cache's memory. Applicant has purports the functionality of the cache in a far too simplistic fashion and also ignores the fact that WinRoute is a complex system that includes a cache only as one of the many features within the system.

Appellant respectfully submits that the Examiner's rejection is without merit. Appellant submits that even assuming *arguendo* the teachings of the cited references could be combined, which Appellant does not admit, the combined teachings of Tso, WinRoute, and Burns would not disclose or suggest combination and/or modification in such a way as to read on all the elements of amended independent claims 1, 8, and 15.

In particular, Tso, WinRoute, and Burns each describe a caching mechanism and thus teach away from deleting information upon delivery. Indeed, the WinRoute "Continue Aborted" feature teaches away from the deletion of information after delivery to a client having issued several requests for it, and is directed to solving a problem of a different nature.

With respect to deleting cached content, Burns teaches the following:

> The content is downloaded from the content provider during the off-peak hours and cached at the local service providers for serving to the subscribers during the ensuing peak time. ***In this manner, the frequently requested content is already present at the local service providers and ready to be served to the subscribers before they actually request it***.
>
> [T]he local service provider 110 also includes a policy manager 128 which defines and administers rules that determine which documents or resources are cached in the cache memory 124. For instance, caching rules might call for caching resources that are routinely requested by many subscribers, but foregoing caching resources that are rarely or infrequently requested. ***The policy rules also coordinate cache maintenance by***

> ***deciding when documents are out-of-date and how these documents are deleted from the cache memory 124***.
>
> Deletion policies are a function of the content itself (e.g., its TTL tags), the subscriber patterns (e.g., how frequently the content is requested), the cost to request newer updated content, and the constraints imposed by capacity limitations of the cache memory.

From the above, Burns clearly teaches caching frequently requested content so that such content is available to subscribers before they request it. Accordingly, Burns fails to teach and indeed teaches away from deleting information at a network node upon delivery of the information to a client. At most, Burns teaches deleting out-of-date documents from cache memory according to certain deletion policies.

Accordingly, even assuming that the teaching of the Tso, WinRoute, and Burns could be combined, such combination of teachings would not motivate one of ordinary skill in the art to delete information at a network node upon delivery of the information to a client as recited in independent claims 1, 8, and 15.

Furthermore, Appellant disagrees with the rationale set forth in the Final Office Action that cache memory would be more efficiently utilized if content were deleted as soon as possible. As taught by Burns, content is cached when such content is routinely requested by many subscribers. If cached content were to be deleted upon delivery to a subscriber, the content would no longer be available to others subscribers before they request it. Accordingly, the modification of cache memory proposed by the Examiner would change the principle of operation of the cited references and render the cited references unsatisfactory for their intended purposes.

Moreover, Appellant takes issue with the Examiner's statements in the Advisory Action. In short, the Examiner argues that because Burns does not imply that cached data

will not be deleted immediately after use, it would be obvious to modify Burns to do so. The Examiner is ignoring the explicit teachings or lack thereof in Burns and is essentially stating that it would be obvious to make a modification which neither disclosed nor suggested so long as there is no explicit or implicit teaching away from such modification. This is clearly improper.

In view of the above, Appellant submits that the Examiner has failed to meet the burden of establishing a *prima facie* case of obviousness with respect to claims 1-18. According to MPEP § 2143, three basic criteria must be met to establish a *prima facie* case of obviousness. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the reference (or references when combined) must teach or suggest all the claim limitations. The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art and not based on applicant's disclosure. *In re Vaeck,* 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991). See MPEP 706.02(j).

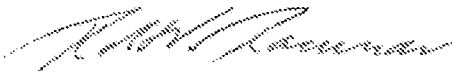
Moreover, all of the teachings of the cited references must be considered, even disclosures that teach away from the claimed invention. *See* MPEP § 2141.02. In addition, the proposed combination cannot render the cited references unsatisfactory for their intended purpose or change the principle of operation of a reference. *See* MPEP § 2143.01, for example. Thus, it is improper to combine references where the references teach away from their combination. *See* MPEP § 2145, for example.

Appellant respectfully submits that Tso, WinRoute, and Burns, whether taken alone or in combination, are insufficient to establish a *prima facie* case of obviousness with respect to independent claims 1-18. Appellant submits claims 1-18 are allowable for at least this reason. Appellant further submits that claims 2-7, 9-14, and 16-18 are allowable by virtue of their dependency, as well as on their own merits.

In view of the foregoing arguments, Appellant respectfully requests the Board to overturn the § 103(a) rejections of claims 1-18.

Respectfully submitted,

KACVINSKY LLC



Robert V. Racunas, Reg. No. 43,027
Under 37 CFR 1.34(a)

Dated: September 28, 2007

4500 Brooktree Road, Suite 102
Wexford, PA 15090
(724) 933-5529

VIII. CLAIMS APPENDIX

1. A method to retrieve information, comprising:

receiving a first request for information from a client at a network node over a first connection;

establishing a second connection to retrieve said information;

detecting that said first connection is terminated prior to retrieval of said information over said second connection;

retrieving said information over said second connection;

receiving a second request at the network node for said information over a third connection;

determining whether said second request matches said first request, including whether said second request is from said client;

sending said information over said third connection in accordance with said determination; and

deleting said information at the network node upon delivery of said information to said client.

2. The method of claim 1, wherein said first request comprises a first source address and a first information address, further comprising storing said information with said first source address and said first information address in an information table prior to receiving said second request.

3. The method of claim 2, wherein said second request comprises a second source address and a second information address, and said determining comprises:

searching said information table to determine whether said second source address matches said first source address; and

determining whether said first information address matches said second information address.

4. The method of claim 3, wherein said source addresses comprise Internet addresses, and said information addresses comprise uniform resource locators.

5. The method of claim 1, wherein said information comprises a hypertext markup language (HTML) file.

6. The method of claim 1, wherein said information comprises an extensible markup language (XML) file.

7. The method of claim 1, further comprising:

receiving a request to terminate said third connection; and

terminating said second and third connections.

8. A method to retrieve information in a network, comprising:

receiving a first request for information from a first network node at a second network node over a first connection;

sending said first request over a second connection to a third network node;

receiving a notice that said first connection is terminated prior to retrieval of said information over said second connection;

receiving said information over said second connection at said second network node;

receiving a second request for said information over a third connection at said second network node;

determining whether said second request matches said first request, including whether said second request is from said first network node;

sending said information over said third connection to said first network node; and

deleting said information at the second network node upon delivery of said information to said first network node.

9. The method of claim 8, wherein said first request comprises a first source address and a first information address, further comprising storing said information with said first source address and said first information address in an information table at said second network node prior to receiving said second request.

10. The method of claim 9, wherein said second request comprises a second source address and a second information address, and said sending comprises:

searching said information table to determine whether said second source address matches said first source address;

determining whether said first information address matches said second information address; and

sending said information in accordance with said determination.

11. The method of claim 9, wherein said source addresses comprise Internet addresses, and said information addresses comprise uniform resource locators.

12. The method of claim 8, wherein said information comprises a hypertext markup language (HTML) file.

13. The method of claim 8, wherein said information comprises an extensible markup language (XML) file.

14. The method of claim 8, further comprising:

receiving a request to terminate said third connection; and

terminating said second and third connections.

15. An article comprising:

a storage medium;

said storage medium including stored instructions that, when executed by a processor, result in receiving a first request for information from a client at a network node over a first connection, establishing a second connection to retrieve said information, detecting that said first connection is terminated prior to retrieval of said information over said second connection, retrieving said information over said second connection, receiving a second request for said information at the network node over a third connection, determining whether said second request matches said first request including whether said second request is from said client, sending said information over said third connection in accordance with said determination, and deleting said information at the network node upon delivery of said information to said client.

16. The article of claim 15, wherein said first request comprises a first source address and a first information address, and the stored instructions, when executed by a processor, further result in storing said information with said first source address and said first information address in an information table prior to receiving said second request.

17. The article of claim 16, wherein said second request comprises a second source address and a second information address, and the stored instructions, when executed by a processor, further result in determining whether said second request matches said first request by searching said information table to determine whether said

second source address matches said first source address, and determining whether said first information address matches said second information address.

18. The article of claim 15, wherein the stored instructions, when executed by a processor, further result receiving a request to terminate said third connection, and terminating said second and third connections.

IX. EVIDENCE APPENDIX

None

X. RELATED PROCEEDINGS APPENDIX

None